# OAK LEAVES

ATOM USERS' GROUP CANADA
Newsletter #8

Thank you all for your patience. Here at last is issue #8 of the
Newsletter. I won't bore you with excuses for the delay.
Sufficient to say that circumstances have conspired to prevent my
providing better service. Similarly it becomes necessary to pass
on the publication to others who will certainly do better than I.
Like many others I bought the Atom to learn about computers, and
I have certainly done that. Thanks to all who have sent in items
for publication over the years. Thanks for all the letters and
phone calls. As you will see from the contents of this issue the
Atom is still alive and well. The Toronto area group is certainly
very vibrant. There is also a move underway to the BBC machine,
hopefully the new publishers will be able to cater for users of
both computers.

John Wood
June 1985

Correction
In NL-#7 the execution address for CALC was given incorrectly.
The correct address should be #A02A.
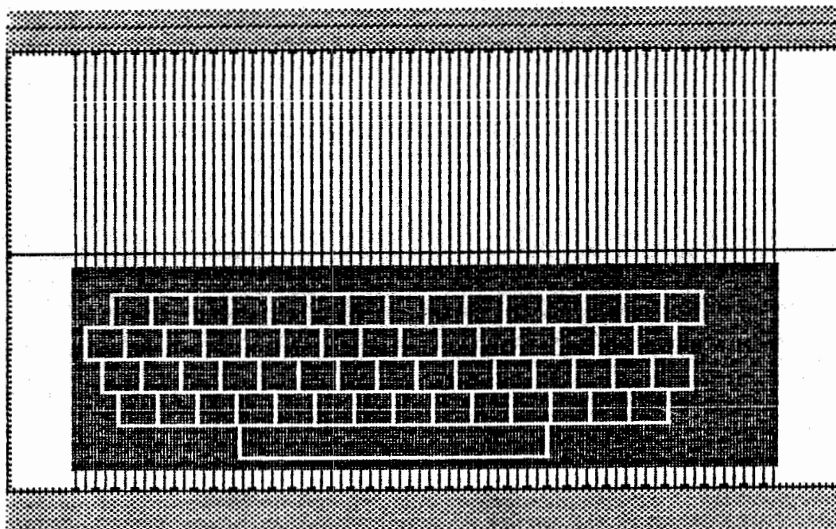
THE BBC MICRO - 6 MONTH USER REPORT

The BBC Micro is the successor to the Acorn Atom, and it has proved to be a very worthy one indeed. The $1500 price tag of the machine marketed in Canada does, however, put it right out of the Atom's price bracket. An alternative is to import a machine direct from England. The author did this in June 1984, and the price of the model B with disc controller, in Canada with duty paid, was $816. (About $650 in May 1985. J.W.) There are other advantages to the English version of the machine. It lacks the internal shielding which was added to comply with the U.S. FCC's requirements on RF emissions, and so is much easier for the experimenter to work on. Also, there is a great deal of information available on both the hardware and software of the English machine. There are enough differences between the two versions to limit the value of this information when applied to the North American version. There are a couple of disadvantages too. You have to modify the internal power supply to run off 110V (a trivial matter of adding a jumper), and the video frame frequency is 50Hz, rather than 60. I was able to modify my Zenith 121 monitor by adding a single capacitor in the vertical oscillator, and it now locks onto both 50Hz and 60Hz signals. (Not needed on mine. J.W.) I haven't tried a TV. A monochrome one might well work, but colour certainly won't. With an 80 column screen, few users would want to use TV anyway.

First impressions of the BBC micro - or "Beeb", as it's called in England - are that it has all the advantages of a very fully expanded Atom built right in, and many, many more. Secondly, with only a very few minor exceptions, all the software works! My machine is a model "B", which is by far the most popular. It has 32K RAM, 32K ROM, a built in controller for up to 4 disc drives, a printer interface, an 8 bit parallel port, a serial port, 4 analog inputs, RGB colour output, and a 4 channel sound generator. The options I've added are the "View" wordprocessor ROM, the speech synthesizer, and the "Viewsheet" spreadsheet ROM. The first two are standard on the North American machine, by the way.

The version of Basic employed is said to be close to Microsoft, but the extensions provided really make it quite unique. Variable names can be of arbitrary length - all the characters are recognized - and can be in upper or lower case. Procedures and Functions aid in writing structured code. In all the time I've had the machine, I've still to use a GOSUB! The Atom-like facilities of Hex arithmetic, easy switchover to assembler language, and the indirection operators "?" and "!" are retained.

The user manual is an extremely informative 519 page tome, which might be a bit daunting for a beginner, but nevertheless only covers the features of the machine available to the Basic programmer. Those who want a more complete description of the operating system should buy the "Advanced User Guide" - another 509 pages! The Disc Operating System, and all the ROMs, each have their own (slimmer!) manuals.

The processor used is a 6502, running at 2MHz, or twice the speed of the Atom. The whole hardware setup lacks the simplicity of the Atom one. The system software is also much more complex. Digging into it with a disassembler is a much more challenging task than on the

Atom, but the Advanced User Guide gives a great deal of the information we had to dig into the Atom ROMs to discover.

One of the most interesting features of the machine is its ability to use a second processor. this should give the product a good few years of life, as new processor technology can be grafted on without scrapping the original machine. Currently available are a second 6502 running at 3MHz, and a Z80 running CPM. Both come with and extra 64K of RAM. The only part of this additional memory which is not available to the user is the 20K or so used for the language software. None is chewed up by CRT buffers, I/O, etc. Apparently, they also have a 16032 second processor on the drawing board, running UNIX. This would be a formidable machine.

If anybody is interested in the Beeb, please feel free to contact me. I have quite a lot of information, and would be glad to have a fellow user in Canada to swop ideas.

Alan Hepburn,
RR#3, Georgetown,   (416) 878 8691
Ont. L7G 4S6.

If you are thinking of ordering your BBC from Britain, you can find a number of suppliers who offer good service on overseas shipments, often accepting phone orders and allowing payment by Visa or Master Card. Twillstar Computers Ltd. Tel:(01) 574-5271, and Watford Electronics Tel:(0923) 50234, have both shown that they can ship goods to be received in Canada within 10 - 15 days of placing an order. The current price for the BBC Micro with disk interface is less than 350 pounds sterling.

There are a number of Acorn RGB vision 111 colour monitors being sold in Canada. Parts Galore on Queen Street in Toronto had some at $450. These are North American standard, but fully compatible the British Beeb. They also work with IBM FC. On the BBC machine set the manual link S31 to "East" to get the correct vertical sync signal, and enter *TV0,1 for a rock steady picture.

DISC CONTROLLER CARDS

Many people now have a disk drive based on the Canadian FDC card. These cards are available from A. G. Ernest, 3435 Cawthra Road, Mississauga, ONT. (416) 277-9649. The price is $35 for a printed circuit board with parts list and guide to assembly.

Some users have noticed an odd problem with various FDC cards (including this one). Occasionally the disk drive will not function when the Atom is first switched on. If it is left on for a few minutes, then switched off and on again, it works fine. The problem seems to lie in an unstability in the 4MHz crystal circuit, which can oscillate at double the correct frequency. The fix is fairly simple if you want to do it. Two 470 ohm resistors (R3 and R4 on the Canadian board) must be changed to 1000 ohm, and a small 22 pF capacitor added between each leg of the 4 MHz crystal and ground. If you have this problem on an Acorn or Control Universal FDC card you can find the 470 ohm resistors connected across pairs of pins on a 7404 or 74LS04 chip.

News from ATOMOTA

ATOMOTA is an Atom owners' group based in the Toronto area dedicated to continuing support for te Atom. Meetings are held on the second Wednesday of each month in the auditorium downstairs in the Downsview Public Library, 2393 Keele St. just north of the 401. The doors open at 7:00 p.m. with the meetings beginning at 7:30. There are no membership dues and all are welcome. A $2 admission charge is collected from those attending to cover the cost of renting the auditorium.

The group meeting on March 13th saw the release of an advanced Operating System and DOS for the Atom, version 1.3. The main features of the system are faster screen-writing (typically 5 times faster than usual on the Atom), Auto-booting of disks on power-up and BREAK, faster keyboard scanning (30 characters per second rather than 10), and a fix for the paged-mode operation that prevents the first page from scrolling off the screen. The main Operating System and the DOS versions 1.3 are available on eproms at $25.00 each, or $10.00 each if you supply the Eproms (a 68764 for the main Operating System and a 2532 for DOS).

At the previous meeting on February 13th, two software packages were released, SYSTEMDISK and ATOMFORTH-1. The SYSTEMDISK is a utility diskette that has a battery of routines fr disk-based Atoms. Included are utilities to rename, copy, boot, dump, list, type, lock or unlock files, format, verify, compact or copy disks, as well as other useful features. Hefty documentation comes with the disk making for a rather powerful and complete package.

The ATOMFORTH-1 package contains a much improved version of ATOMFORTH which fully implements the standard fig-FORTH model as well as many features of FORTH-79 and FORTH-83. Also included is a substantial package of utilities which make this version very powerful and flexible. Given enough memory, the application area of the dictionary can contain almost 16K of compiled FORTH words. The EDITOR screens have been compacted, thus freeing disk space for other utilities. A new word, INDEX, ha been added to the EDITOR which allows the display of line 0 (usally reserved for the screen title) of any group of contiguous screens. The COPY utility has been modified to allow the copying and editing of a maximum of 23 screens at a time instead of 5. A virtual I/O system has been developed to allow manipulation of blocks of data and FORTH words in RAM instead of disk, thus speeding up data handling. The sequential file commands of BASIC have been duplicated in ATOMFORTH-1 to enable complete control of disk files from within FORTH. And a new utiltity will allow the user to create a custom version of FORTH, incorporating in the core dictionary any desired set of words and utilities.

SYSTEMDISK and ATOMFORTH-1 are available from ATOMOTA at $25.00 each, and have both proven to be real bargains.

A final note from ATOMOTA. We hear with great sorrow that a number of Atom owners are giving up on the Atom and selling their machines. It seems as though lack of support and lack of hardware and software developments is at the root of user disatisfaction. If you are thinking of abandoning the Atom, please consider the support that ATOMOTA can supply. These are some of the things we presently offer:

-comprehensive repair service (we haven't found a problem we couldn't fix)
-a source of parts for the Atom including single and double sided drives, chips, memory boards, disk controller cards etc.
-loads of software
-technical advice
-programming advice

And soon we will be offering:

-80 column screen
-serial card and direct-connect modem
-screen noise elimination circuit
-a disk-based Database that puts all others for the Atom to shame
-an analog-to-digital card with 8 channels to handle joystick, mouse, graphics tablet or any other analog device -a graphics package that allows easy operator creation of complex forms and fill-in of areas -a bigger and better word processor based on Wordpak allowing for the creation of headers and footers -an honest-to-goodness DOS manual with complete disassembly and documentation of DOS -a background print spooler that allows for the printing of a file while doing other processing

But if you are still resolved to sell your Atom, give us a call. We have users that are looking for second machines.

ATOMOTA (Atom Users of the Toronto Area)
c/o Richard Bales, President
97 Golfview Avenue
Toronto, Ontario, M4E 2K3   Tel: (416)-690-6597

WANTED/FOR SALE

Now that I've installed RAM in #A000-#AFFF. I don't need my Toolbox ROM or Wordpack ROM. not to mention the two way ROM selector board and switch. You can have the whole shooting match for $55, including instructions and postage.
John Lasruk, 448 Runnymede Rd, Toronto, Ont. M6S 2Z1

Chris Hanks is looking for a ROAM board. Call him: (416) 499-1834
Or write: #1214, 20 Chichester Place, Agincourt, ONT. MIT 1G6

```
  10REM ***satellite*** by Reg Peers
  60FDIM%FF(24);DIMH(15),I(1);@=0
 110P.$12"THIS IS A PROGRAM TO CALCULATE"'"THE LOOK ANGLE "
 160P."FROM ANYWHERE ON THE NORTH-AMERICAN CONTINENT TO"'
 210P."ANY GEO-SYNCHRONOUS SATELLITE"'"ABOVE IT"'
 260F.N=1TO100;WAIT;N.;P.'''"DO YOU WISH TO"'
 310P." #1-USE THE EARTH-STATION"'"      CO-ORDINATES BUILT IN"'
 360P." #2-CREATE NEW ONES"'
 410IN.Z
 460IF Z=2 G.560
 510A=44;B=15;C=79;D=11;G.760
 560P.$12"EARTH STATION LATITUDE ?"'
 610IN."DEGREES"A,"MINUTES"B;P.'
 660P."EARTH STATION LONGITUDE ?"'
 710IN."DEGREES"C,"MINUTES"D;P.'
 760P."THE FOLLOWING IS A LIST OF"'"THE SATELLITES AVAILABLE"''
 810F.N=1TO50;WAIT;N.;P.$12
 860P."#1-WESTAR 1     --    99.0 DEGREES"'
 910P."#2-WESTAR 11    --   123.5 DEGREES"'
 960P."#3-WESTAR 111   --    91.0 DEGREES"'
1010P."#4-COMSTAR 1    --   128.0 DEGREES"'
1060P."#5-COMSTAR 11   --    95.0 DEGREES"'
1110P."#6-ANIK 1       --   104.0 DEGREES"'
1160P."#7-ANIK 11      --   109.0 DEGREES"'
1210P."#8-ANIK 111     --   114.0 DEGREES"'
1260P."#9-SATCOM 1     --   135.0 DEGREES"'
1310P."#10-SATCOM 11 --   119.0 DEGREES"'
1360P."#11-GALAXY      --  143.0 DEGREES"'''
1410F.N=1TO50;WAIT;N.
1460IN."PICK THE SATELLITE OF YOUR"'"CHOICE BY NUMBER (1-11)"G
1510IF G=1 $H="WESTAR 1";%E=99
1560IF G=2 $H="WESTAR 11";%E=123.5
1610IF G=3 $H="WESTAR 111";%E=91
1660IF G=4 $H="COMSTAR 1";%E=128
1710IF G=5 $H="COMSTAR 11";%E=95
1760IF G=6 $H="ANIK 1";%E=104
1810IF G=7 $H="ANIK 11";%E=109
1860IF G=8 $H="ANIK 111";%E=114
1910IF G=9 $H="SATCOM 1";%E=135
1960IF G=10 $H="SATCOM 11";%E=112
2010IF G=11 $H="GALAXY";%E=143
2060IN."DO YOU WANT A HARD COPY (Y/N)"$I
2110IF $I="Y"P.$2
2160%FF(0)=%(A+(B/60));REM EARTH STATION LATITUDE (DECIMAL)
2210%FF(1)=RAD %FF(0);REM CONVERT LATITUDE TO RADIANS
2260%FF(2)=%(C+(D/60));REM EARTH STATION LONGITUDE (DECIMAL)
2310%FF(3)=RAD %FF(2);REM CONVERT LONGITUDE TO RADIANS
2360%FF(4)=%E-%FF(2)
2410REM DIFFERENCE IN LONGITUDE BETWEEN EARTH STATION & SAT.
2460%FF(5)=RAD %FF(4);REM CONVERT THIS DIFFERENCE TO RADIANS
2510%FF(6)=COS%FF(1);REM COSINE OF EARTH STATION LATITUDE
2560%FF(7)=COS%FF(5);REM COSINE OF THE DIFFERENCE IN LONGITUDE
2610%FF(8)=%FF(6)*%FF(7)
2660%FF(9)=SQR(1-%FF(8)^2)
2710%FF(10)=ATN%(%FF(9)/%FF(8))
2760REM ANGLE BETWEEN EARTH STATION & SATELLITE IN RADIANS
2810%FF(11)=TAN%FF(10)
2860%FF(12)=TAN%FF(1)
```

```
2910%FF(13)=%FF(12)/%FF(11)
2960%FF(14)=SQR%(1-%FF(13)^2)
3010%FF(15)=ATN%(%FF(14)/%FF(13))
3060%FF(16)=DEG %FF(15)
3110%FF(16)=%FF(16)+180;REM AZIMUTH BEARING FROM TRUE NORTH
3160%FF(17)=.15116235
3210%FF(18)=COS%FF(10)
3260%FF(19)=%FF(18)-%FF(17)
3310%FF(20)=SIN%FF(10)
3360%FF(21)=%FF(19)/%FF(20)
3410%FF(22)=ATN%FF(21)
3460%FF(23)=DEG %FF(22)
3510P.$12'''"THE REQUIRED LOOK ANGLE"'"FOR "$H" AT"'
3560P.%E;P." DEGREES LONGITUDE IS"'''
3610P."AZIMUTH = ";FP.%FF(16);P." DEGREES"'
3660P."REFERENCED TO TRUE NORTH"''
3710P."ELEVATION = ";FP.%FF(23);P." DEGREES"'
3760P."ABOVE THE HORIZON"'" "'$3
3810P."PRESS ANY KEY TO CONTINUE";LI.#FFE3
3860G.810
3910E.
```

```
                    ROUNDING ROUTINE

   10 REM ATOM USER/ATOM FORUM
   11 REM AUGUST
   20 DO;FIN."F.P. NUMBER"%N;GOS.z;P.';U.0
  100E.
1000zP=#8200;STR%N,P;?(P+20)=13;IF?(P+10)<>CH"E"G.1020
1010 F.Z=(P+10)TO(P+14);Z?10=?Z;N.
1020 Z=P+10;DO;?Z=13;Z=Z-1;U. ?Z<>CH"0";IF ?Z=CH"=";?Z=13
1030 P.$P$(P+20);R.
```

MACHINE CODE PATCH TO GALAXIAN FOR JOYSTICKS

#31A7	AD	1	B8	29	4	D0	A	AD
#31AF	28	3	C9	E8	F0	3	EE	28
#31B7	3	AD	1	B8	29	1	9	2
#31BF	EA	EA	EA	AD	1	B8	29	10
#31C7	D0	A	AD	28	3	C9	1	F0
#31CF	3	CE	28	3	60	20	A7	31
#31D7	AD	43	3	8D	56	3		
#3210	AD	44	3	F0	B	4C	F1	36
#3218	A9	0	8D	85	3	4C	10	33
#3220	AD	6A	3	D0	55	AD	1	B8
#3228	29	1	D0	EC	AD	85	3	F0
#3230	3	4C	10	33	18	AD	3C	3

Scott McLellan has provided a fix to allow another popular game to be
played using a joystick. For details on how to connect the joystick
see Oakleaves issue #6 (February 1984)

6.   The floating-point workspace

```
#2800-#2887    Storage for F.P. variables %@@ to %ZZ. See p.165, AT&P
#2888-#28FF    Used to hold array indexes, as well as for temp. use
```

7. Useful ROM Addresses

```
#C2AD    Entry point of NEW statement
#C2B2    Normal entry point for BASIC
#C325    Entry point for LET statement
#C334    Entry point for PRINT command
#C3B2    Entry point for LINK command.  (#05),Y should point to
         expression for link address.
#C3D0    Transfer no. in #16,X; #25,X; #34,X; #43,X to 0,Y; 1,Y;
         2,Y; and 3,Y
#C3EE    Entry point for ! operator (on L.H. side)
#C40F    Entry point for execution of O.S. commands
#C424    Checks for F.P. ROM  Returns with C=0 if ROM not present
         otherwise, C=1
#C434    Checks for variable in (#05),Y.  C=0 if no variable at
         beginning of string; otherwise C=1
#C46A    Convert ASCII digits at (#05),Y into a hex number at
         #16,X; #25,X; #34,X; #43,X.  C=0 if no number present
#C566    Entry point for IF statement
#C575    Entry point for REM statement
#C589    Print 4-byte hex number in #16, #25, #34, #43 as an
         ASCII string (in decimal).  Uses field size @
#C75B    Entry point for = operator
#C764    Entry point for <= operator
#C76D    Entry point for <> operator
#C774    Entry point for < operator
#C77B    Entry point for >= operator
#C782    Entry point for > operator
#C78B    Interpret expression starting at (#05),Y and store result
         in #14,X; #23,X; #32,X; #41,X. Note that X should be >=2
#C902    Entry point of ABS function
#C90A    Entry point for # operator (hex number symbol)
#C973    Entry point for TOP function
#C97A    Entry point for COUNT function
#C986    Random number seed.  A new random number is generated and
         stored in #16,X; #25,X; #34,X; #43,X. A=0 upon return.
#C9BD    Entry point for LEN function
#C9D2    Entry point for CH function
#C9D8    Normal entry point for the BRK service routine.
#CA2C    Evaluate both sides of an expression in the form
         X=<expression>.  The value of X IS changed by the routine
#CA37    Set the variable denoted by Y (Y=0 for @, Y=26 for Z)
         to equal #17,X; #26,X; #35,X; #43,X.
#CA4C    Print a character and adjust the column counter.
```

```
#CA51    Entry point for the LIST command.
#CACD    Entry point for NEXT statement
#CB57    Entry point for FOR statement
#CBD2    Entry point for GOSUB statement
#CBEC    Entry point for RETURN statement
#CC05    Entry point for GOTO statement
#CC81    Entry point for INPUT statement.  (#05),Y should point to
         the following: the message string, in quotes, the list of
         variables to be inputted, and a <CR> code.  Unfortunately,
         this routine dumps you back into BASIC, so have the line
         finish off with ';LINK #nnnn<CR>' to get back into ML.
#CCD2    Entry point for UNTIL statement
#CCF0    Entry point for DO statement
#CD0F    Snuff in an input line starting at #0100 (maximum 64
         characters long), printing the character in A upon entry
         as a prompt.  Makes use of the usual control codes.
#CD98    entry point for END statement.  Returns to BASIC (not to
         user routine)
#CD9B    As above but does not perform the usual syntax checks.
#CEED    Entry point for LOAD command
#CF0A    Entry point for SAVE command
#CF28    Entry point for EXT function
#CF29    Entry point for PTR function
#CF47    Entry point for PTR command
#CF5B    Entry point for BGET function
#CF66    Entry point for GET function
#CF8F    Entry point for BPUT command
#CF95    Entry point for PUT command
#CFA6    Entry point for FIN function
#CFA7    Entry point for FOUT function
#CFB6    Entry point for SHUT command
#CFC5    Entry point for SPUT command
#CFE3    Entry point for SGET command
#E000    Entry point into the DOS
#E00D    As for #E016 below, except it prints the word 'DISK ' first
#E016    Prints the message string following the calling statement.
         Execution is returned upon the first negative or zero (break)
         code. Eg.:
         JSR #E016  - call subroutine
         ...........  - ASCII data to be printed out
         NOP          - This instruction is executed, since the
         opcode for a nop is negaive (ie. greater than 127)
#E0AB    Print 'Name?  Error' follwed by a BRK instruction (error)
#E0FA    'LSR A' instruction chain.  A call to #E0FA will perform
         5 'LSR A's, a call to #E0FE will perform 1, and so on.
#E100    'INY' chain.  A call to #E100 will perform 8 'INY's, and
         a call to #E107 will perform 1 'INY', and so on.
#E109    'DEY' chain.  A call to #E109 will perform 8 'DEY's, and
         a call to #E110 will perform 1 'DEY', and so on.
#E154    Print a 'FILE?  ERROR' followed by a BRK.
#E1AA    Print a 'PROT  ERROR' followed by a BRK.
#E1B2    Entry point for *INFO command.  Filename should be in
         #0100,Y
#E1BF    Print information on file with filename at #2008,Y
```

```
#E226    Wait until current disk access is completed.
#E231    Entry point for *DIR command.  #0100,Y should point to
         either a <cr> or a <drivenumber><cr> sequence.
#E237    Entry point for *CAT command.  Requirements as for *DIR
#E3E5    Entry point for command line interpreter.
#E14A    Entry point for *DELETE command.  #0100,Y should point to
         a <filename> followed by a <cr>
#E43F    Set drive number.  The drive number should be in A.
#E44F    Print 'DRIVE? ERROR', followed by a BRK.
#E459    *NOMON entry point.  #0100,Y should point to <cr>
#E45B    *NOMON entry point.  Requrements same as for *MON
#E465    Entry point for *LOAD.  Requirements as per *DELETE
#E477    O.S.LOAD. Entry point
#E4C5    'Utility Program' entry point.  #0100,Y should point to
         the name of the program, followed by a <cr> code.
#E50A    *RUN entry point.  Requirements as per *DELETE
#E519    Entry point for *EXEC command.  Requirements as per *DELETE
#E547    Entry point for *SPOOL command.  Requirements as per *EXEC
#E565    Entry point for *GO command.  #0100,Y points to a hex #
#E575    *SET entry point.  A  should contain the new qualifier.
#E578    *TITLE entry point.  #0100,Y should point to a char. string
#E599    *LOCK entry point.  Requirements as per *DELETE
#E59A    *UNLOCK entry point.  Requirements as per *DELETE
#E5AF    *USE entry point.  #0100,y should point to <qual><cr>.
#E5D3    Prints 'SYNTAX? ERROR', followed by a BRK.
#E5DE    Prints 'FULL  ERROR', followed by a BRK.
#E5E6    *SAVE entry point.  #0100,Y should point to <filename>
         <start> <end> <execution><cr>. Execution address optional.
#E613    O.S.SAVE. Entry point.  See AT & P for parameters.
#E6B8    *VDU entry point.
#E731    Copy file directory into RAM off disk.
#E87B    Start of NMI routine
#E89C    *SHUT entry point
#E89E    O.S.SHUT entry point
#E953    O.S.FIND entry point
#EA61    O.S.RDAR entry point
#EAF0    O.S.BGET entry point
#EBBC    O.S.BPUT entry point
#ECA0    O.S.STAR entry point
#F0AE    DIM statement entry point.
#F2A1    *** START OF THE MIGHTY ASSEMBLER ***
#F67B    Entry point for CLEAR command
#F6C2    Entry point for CLEAR 0
#F6E2    Entry point for Mode 0 graphics
#F73B    Entry point for Mode 1 graphics
#F754    Entry point for Mode 2 graphics
#f76D    Entry point for Mode 3 graphics
#F7AA    Entry point for Mode 4 graphics
#F7D1    Message printing routine.  Exactly identical to #E016
         except control IS NOT transferred on a 00 code.
#F802    Print A as two hex digits
#F86C    Print 'NAME  ERROR', followed by a BRK.
#F87E    Convert ASCII digit (0-9,A-F) in A to hexadecimal.
         C=1 if no number present
```

```
#F893    Convert ASCII hex digits at #0100,Y into a hex no. at
         #00,X and #01,X.   Z=1 if no number present.
#F8EF    Command line interpreter.
#F926    Print 'COM? ERROR' followed by a BRK.
#F955    *FLOAD entry point
#F958    *LOAD entry point
#F96E    O.S.LOAD entry point.
#FA19    *NOMON entry point
#FA1A    *MON entry point
#FA20    *RUN entry point
#FA2A    *CAT entry point
#FA7D    Print 'SYN?  ERROR' followed by a BRK.
#FABB    *SAVE entry point
#FAE5    O.S.SAVE entry point
#FB7D    Wait 2.0 Seconds
#FB83    Wait X/60 seconds
#FBEE    O.S.BGET entry point
#FC38    O.S.FIND entry point (print messages)
#FC7C    O.S.BPUT entry point
#FD1A    Bleep speaker
#FD44    Invert char. at current cursor position
#FE52    O.S.WRCH entry point
#FE55    As above, but does not attempt to write to printer
#FE5C    Actual screen-writing routine
#FE66    Wait until next VDU flyback period
#FE6B    Wait until flyback (can be current period)
#FE71    Scan keyboard.  Key number in Y, C=1 if no keypress
#FE94    O.S.RDCH entry point.  Returns with ASCII value in A
#FEFB    Write ASCII char. in A to printer if enabled.
#FF3F    Start of RESET routine
#FFB2    IRQ/BRK routine
#FFC7    NMI routine.  Transfers execution to (#0200)
#FFCB    O.S.SHUT routine
#FFCE    O.S.FIND routine
#FFD1    O.S.BPUT routine
#FFD4    O.S.BGET routine
#FFD7    O.S.STAR routine
#FFDA    O.S.RDAR routine
#FFDD    O.S.SAVE routine
#FFE0    O.S.LOAD routine
#FFE3    O.S.RDCH routine
#FFE6    O.S.ECHO routine
#FFED    O.S.CRLF routine
#FFF4    O.S.WRCH routine
#FFF7    O.S.CLI  routine
#FFFA    NMI vector - set to #FFC7
#FFFC    RESET vector-set to #FF3F
#FFFE    IRQ/BRQ vector-set to #FFB2
```

CAUTION!!! Memory locations #00 - #06 MUST be preserved before
 linking up with the BASIC ROM subroutines!!!

OAK LEAVES

ATOM USERS' GROUP CANADA (1985)

This issue of OAK LEAVES is the last one to be published by John Wood, but don't despair your newsletter will continue! Arrangements have been made for OAK LEAVES to be maintained under the direction of a recently formed publishing committee. We will publish quarterly issues of OAK LEAVES commencing in September, with subsequent issues appearing in December, March, and June.

The new editor of your newsletter will be John Lasruk. Membership, circulation, and administrative duties will be handled by Dave Anderson. The third member of the triumverate (computerese for a group of three jointly excercising supreme power), Luigi Bianchi, will handle the correspondence, the accolades, and should there ever be any, the complaints.

A new format for the newsletter is under development, but we will of course continue to provide you with interesting articles on programming, software developments, hardware modifications, system upgrades, etc. If you have any suggestions regarding content, style, format, or would like to contribute an article for a future issue please write and let us know; we want to hear from you.

The ever increasing effects of inflation, particularly postal rate increases, together with some degree of uncertainty regarding the magnitude of our circulation (we're talking 0.1K or thereabouts), have unfortunately forced us to raise the annual subscription rate to $15.00. In order to help us in our attempt to give you the best service possible we ask that you complete and mail the subscription renewal form, included elsewhere in this issue, now. This will ensure that you are on our mailing list for receipt of the four 1985/86 issues of OAK LEAVES.

Suggestions regarding newsletter content or articles for publication should be addressed to: John Lasruk, 448 Runnymede Road, Toronto, Ontario, M6S 2Z1 Tel:(416)-767-7121

Subscription renewals together with cheques or money orders made payable to OAK LEAVES should be sent to: Dave Anderson, 47 Ferrah Street, Unionville, Ontario, L3R 1N6 Tel:(416)-477-3279

Questions about available software and hardware, local user's groups, comments, complaints, etc., may be sent to: Luigi Bianchi, 58 Glenrose Ave., Toronto, Ontario, M4T 1K4 Tel:(416) 487-7645

So don't delay, do it today, and together we'll ensure that the ATOM LIVES!
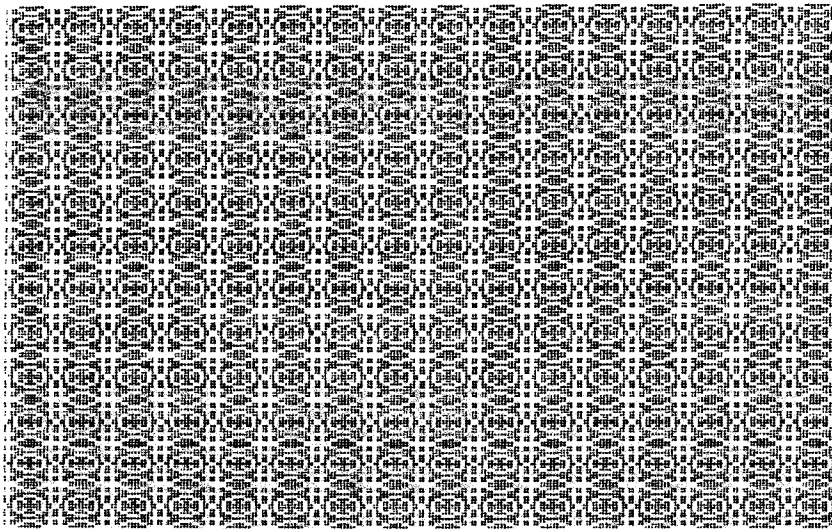
PATTERN MAKER
by John Lasruk

This little routine might appeal to those of you who have a monitor or can dump graphics 4 to a printer. When you RUN it, you will be confronted with a rectangle, inside of which is a little cursor. This cursor may be moved around using the W, A, S, and Z keys. The <space> bar will plant a dot at the current position, making the cursor disappear, though it will reappear as you move it along. This will be familiar to owners of the SOFT VDU program from Acorn. When the @ key is pressed, the designed character is flipped and flopped to make a 16x16 pixel pattern, which is subsequently run repeatedly onto the GR4 screen to make a larger pattern, similar, perhaps, to the one on this page. No, this routine will not guide missiles or improve the profitability of a business, but what the heck...

```
 10 DIMP-1
 20 P.$21;[JSR#FE94;STA#90;RTS;];P.$6
 30 CLEAR1;MOVE7,47;DRAW16,47;DRAW16,38;DRAW7,38;DRAW7,47
 40 X=12;Y=42;Z=#90;PLOT13,X,Y
 50 DO LINK TOP;PLOT14,X,Y
 60 X=X-(?Z=65)-(X>25)+(?Z=83)+(X<0)
 70 Y=Y-(?Z=90)-(Y>48)+(?Z=87)+(Y<30)
 80 IF?Z=32G.100
 90 PLOT14,X,Y
100 UNTIL?Z=64;PLOT14,X,Y;L=#8191
110 MOVE7,47;PLOT7,16,47;PLOT7,16,38;PLOT7,7,38;PLOT7,7,47
120 F.C=#8181TO#8111S.-16
130 ?L=?C;L=L+16;N.
140 F.C=#8111TO#8201S.16;Q=128;R=1
150 DO IF?C&Q=Q C?1=C?1ÐR
160 Q=Q/2;R=R*2;U.R=256
170 NEXT;S=#2800
180 F.C=#8111TO#8201S.16;!S=!C;S=S+2;N.
190 F.T=1TO120;WAIT;N.;CLEAR4
200 F.J=#8000TO#97FFS.#200
210 F.K=J TO J+31S.2;S=#2800
220 F.L=K TO K+480S.32;!L=!S;S=S+2
230 N.;N.;N.;END
```

CUMULATIVE INDEX